

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Original) A method for rendering an assembly of a first object and a second object on a user-interface of a device, the device being either of a first type or of a second type, the first and second objects presenting data of an application, the method comprising:

providing an interpreter specific for an application specification language used to write the application;

storing the interpreter in the device;

receiving an application specification document by the device, the application specification document having a statement with an indication to render the first and second objects in the assembly;

interpreting the statement of the application specification document using the interpreter to identify a presentation pattern for the assembly from predefined first and second presentation patterns according to the type of the device; and

rendering the assembly of the first and second objects on the user-interface according to the presentation pattern identified during the interpreting of the statement.

2. (Original) The method of claim 1, further comprising:

simulating the rendering of the assembly by a pre-viewer component of a workbench used in a development computer.

3. (Original) The method of claim 1, further comprising:

storing the predefined presentation patterns by the interpreter.

4. (Original) The method of claim 1, wherein in the rendering step, the first object and the second objects are rendered according to the presentation pattern and to a predefined hierarchy pattern.

5. (Original) The method of claim 1, wherein the presentation pattern is as a display pattern, wherein the objects are rendered to the user-interface being a screen, and wherein the presentation pattern is identified according to the size of the screen.

6. (Original) The method of claim 1, wherein in the rendering step, the presentation pattern is an audio pattern.

7. (Original) The method of claim 1, wherein receiving an application specification document by the device includes:

automatically receiving an application specification document by the device from a server computer when application data changes on the server computer.

8. (Currently Amended) A computer-program product to visually render a first object and a second object in an assembly on screen of a computing device, the objects presenting data of an application on a computer that is at least temporarily coupled to the computing device, the device being either of a first type or of a second type, the computer-program product embodied in a computer-readable storage medium and having instructions that cause a processor of a computing device to:

provide an interpreter specific for an application specification language used to write the application;

store the interpreter in the computing device;

receive an application specification document from the computer, the application specification document having a statement with an indication to render the first and second objects in the assembly;

interpret the statement of the application specification document using the interpreter to identify a visual presentation pattern for the assembly from predefined first and second visual presentation patterns according to the type of the device; and

render the assembly of the first and second objects on the screen according to the visual presentation pattern identified in the interpreting step.

9. (Currently Amended) A method for creating an application system operating with a computing device and rendering an assembly on a user interface of the device, the method comprising:

defining a user-interface model;

defining an application specification document by a meta-language;

customizing a workbench component that identifies constraints on the validity of the application specification document;

defining layout themes for the computing device;

realizing the user-interface model in an interpreter component; and

realizing the layout-themes in the interpreter component; and

rendering a first object and a second object on the user interface of the device using the user interface model according to one of the layout themes for the device after receiving, at the device, the application specification document, wherein the application specification document includes a statement with an indication to render the first and second objects in the assembly.

10. (Original) The method of claim 9, wherein defining a user-interface model includes:
determining the types of tiles and the functionality of tiles, the tiles being elements of the user-interface model;

determining relationships between the tiles in an assembly; and

determining a navigation state and the required user operations on the navigation state.

11. (Original) The method of claim 10, wherein defining an application specification document by a meta-language includes:

- defining specifications to the types of tiles;
- defining attributes to express properties of the tiles; and
- defining attributes in the navigation state.

12. (Original) The method of claim 11, wherein defining layout themes for the computing device includes:

- defining a representation on the output media of device for each element of the user-interface model; and
- defining the user-interface model for each operation of the user-interface model.

13. (Original) The method of claim 12, wherein realizing the user-interface model in an interpreter component includes:

- creating models to specify the tiles and the assembly;
- implementing constructors to create user-interface instances from the application specification document; and
- implementing the user-interface instances from the models in a computer programming language.

14. (Original) The method of claim 13, wherein realizing the layout-themes in the interpreter component includes:

- implementing each layout-theme as a layout handler; and
- obtaining a selection of the layout-theme by a developer and forwarding the selection to the interpreter component.